

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Application No.: 10/623,406 § Examiner: Lu, Charles Edward  
Filed: July 18, 2003 § Group/Art Unit: 2161  
Inventors: § Atty. Dkt. No: 5760-12400  
    HanCheng Hsiung  
    Xinyi David Lai  
Title: REFRESH MECHANISM  
        FOR DATABASES

## **REPLY BRIEF**

## **Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir/Madam:

Further to the Examiner's Answer of October 8, 2008, Appellants present this Reply Brief, and respectfully requests that this Reply Brief be considered by the Board of Patent Appeals and Interferences.

## **REPLY TO EXAMINER'S ANSWER**

### **First and Second Grounds of Rejection**

According to the Examiner's Answer, pp. 9-10, both the 112 rejections have been withdrawn.

### **Third Ground of Rejection**

Claims 1, 2, 5-10, 13-16 and 19-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kampe et al. (U.S. Publication 2002/0032883) (hereinafter “Kampe”) in view of Raman et al. (U.S. Publication 2003/0217119) (hereinafter “Raman”). Appellant respectfully traverses this rejection for the following reasons.

- 1. The Examiner has failed to establish a *prima facie* rejection because Kampe in view of Raman fails to teach or suggest loading new data to the database clone, wherein said loading updates the storage checkpoint, wherein the new data is new with respect to the production database.**

The Examiner initially addresses Appellants' assertions regarding “databases” (which, as already noted, appears nowhere in Kampe). Appellants respectfully submit that the paragraph indicated by the Examiner (Appellants' Brief, page 12), while briefly noting the deficiency of Kampe regarding databases, provides an overview of the deficiencies of the Examiner's current rejection and more particularly addresses the combination of Kampe and Raman in general. However, to address the Examiner's assertion that “Kampe suggests a database because Kampe is drawn to a storage system that stores, access, and replicates data”, citing paragraphs [0014], [0015], [0042], and [0065], Appellants respectfully submit that the mere replication of data does not teach or suggest a database of the claims. More

specifically, Appellants disagree that any reference that teaches storage and replication of data inherently refers to databases, which have a more specific meaning in the art.

Next, the Examiner withdraws his previous interpretation of the claimed “new data” as “data that is current, recent, or fresh.” The Examiner then goes on to assert that pages 13 and 14 of the Appeal Brief were drawn towards this line of reasoning and begins to address those arguments presented on page 14 of the Appeal Brief which were more specifically drawn to the interpretation of “new data” as “different data” or “new data that did not previously exist”. However, Appellants respectfully submit that those arguments before page 14 still apply, at least in part, to the Examiner’s current interpretation, and should not be ignored. For example, the following two paragraphs (which are substantially the same as those presented on pages 12 and 13 of the Appeal Brief) addresses how the “high availability” of Kampe actually teaches away from the “load[ing] new data to the database clone” recited in the claims:

With respect to the referenced feature, the Examiner relies on Kampe. First, Appellants note that Kampe teaches a system for maintaining replicas of a primary component on nodes (via checkpointing) for achieving high availability in a cluster (see paragraph [0013]). Thus, Kampe is particularly directed towards achieving high availability of a component (in the case of failover) in a cluster rather than loading new data into a production database. For example, in Kampe, paragraph [0064] (which the Examiner relies on for switching from the previous file system data of the production database to the storage checkpoint to be the file system data for the production database), various failover scenarios are discussed where a primary component PC1 fails and the checkpoint is used to recreate the last consistent state of the primary component (either on the first node or on a second node). Thus, in Kampe, checkpoints are used to maintain the current state (or information for recovery) of a component on the first node, and are not used to load new data (new with respect to the production database) into a production database using a database clone. Said another way, Kampe is particularly devoted to loading current state data to the checkpoint. Thus, the checkpoint data is either old with respect to the current state (i.e., the checkpoint does not have the same data and the data

is out of date) or up to date with respect to the current state (i.e., the checkpoint and the component have same data). Under no condition is the checkpoint data (that is loaded into the database) ever new with respect to the current state (i.e., not the same data and the checkpoint has new data that is not present in the primary component). Thus, Kampe actually **teaches the opposite** from this limitation as Kampe is devoted to replication of data to a checkpoint rather than loading new data into a database clone to update the production database. Therefore, the Examiner has failed to establish a *prima facie* rejection.

Appellants note that the Examiner specifically cites “(504)” for this limitation without any further explanation. 504 is a block in Figure 4A of Kampe which states “continuously update checkpoint”. Thus, 504 relates to maintaining the checkpoint with respect to the current state of the primary component. As argued above, there is no indication in this Figure or anywhere else (in Kampe or Raman) which relates to loading new data to the database clone, wherein the new data is new with respect to the production database. Using checkpoints to maintain a current state (or backup of a current state) does not relate to loading new information into a database clone (or checkpoint) which is new with respect to the production database (or, using the Examiner’s citation, the primary component of Kampe). Thus, claim 1 relates to loading data, which did not previously exist in the production database, into a database clone / checkpoint, and then using the file system data of the checkpoint as the file system data of the production database (thereby loading the new data into the production database). Neither Kampe nor Raman (singly or in combination) teach these features of claim 1. Instead, as indicated above, Kampe maintains a current state of the primary component for failover purposes and does not load new information with respect to the primary component. As indicated above, Appellants further note that loading new data (as recited in Appellants’ claims) into the checkpoints of Kampe would **be counter to** the purpose of high availability in a cluster since the replica / checkpoint would no longer represent the current state of the primary component. Thus, the cited references do not teach or suggest **loading new data to the database clone, wherein said load updates the**

storage checkpoint, wherein the new data is new with respect to the production database. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In the Examiner's Answer, the Examiner does not substantively address the above arguments and only addresses Appellants' arguments which begin on page 14. More specifically, Appellants had argued that the data characteristics of the checkpoint of Kampe (e.g., format states, control blocks, and attributes) do not correspond to "loading new data to the database clone" which updates the storage checkpoint (which is then later used as the file system data for the production database). On this point, the Examiner specifically asserts:

The various data must be "loaded" (i.e., written to a storage device) as claimed, in order to manage the checkpoint. Specifically, the replica in Kampe contains a control block (paragraph 0084) which is a block of data in which new data is written (paragraph 0085, "crcs\_cb\_pwrite")... The examiner respectfully disagrees at least because Appellant is relying on limitations that are not claimed. Specifically, the claims do not require that the loaded new data becomes production data, or is used after failover.

Appellants respectfully submit that the claims specifically require that new data is loaded to the database clone. As one of skill in the art understands, *loading data to a database* has a specific meaning, and is not simply "[writing] to a storage device." More specifically, loading data to the database clone (where that data is later used to be the file system data of the production database) means that the new data becomes production data. Thus, the mere writing of checkpoint attribute data does not constitute "loading new data to the database clone" as required by the claims. Thus, Appellants' arguments (both current and previously submitted) are supported by the clear language of the claims and the Examiner has failed to establish a *prima facie* rejection..

Moreover, the Examiner's assertion, "or is used after failover" has nothing to do with Appellants' claims. As specifically recited in the claims, the database clone is used to add new data – no failover is mentioned or required. The Examiner appears to have equated Kampe's failover teachings with Appellants' claims, which as has already been argued, are simply not equivalent. Thus, as already noted, Kampe is specifically directed towards maintaining identical data and corresponding **teaches away** from loading new

data as required by the claims. Therefore, the Examiner has failed to establish a *prima facie* rejection.

**2. The Examiner has further failed to establish a *prima facie* rejection because the Examiner failed to provide a proper reason to combine to the two references.**

The Examiner asserts it would have been obvious to modify Kampe, “to improve accessibility for file system data, as known to one of ordinary skill in the art and taught by Raman (para. 0049)”. The Examiner has not provided any reason to make the specific combination and instead has merely provided a portion of Raman which describes a method for providing copies of consistent file systems with concurrent read-write updating of the file system. There is no indication in this section (or in the Examiner’s provided reason) as to why the maintaining of file systems as taught by Raman applies to the “primary component” of Kampe. More specifically, the cited portion does not relate to a “production database”, the accessibility of a production database during the load of new data, or any reason to include such features in Kampe as alleged by the Examiner. Instead, Raman teaches that updates may be made over an IP network, and that concurrent read-only access to the remote copies is available. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response to the above arguments, the Examiner reiterates his assertion regarding paragraph [0049] and asserts that the reason to combine “to improve accessibility for data in a file system” “is not only found in the above paragraph of Raman, but is also known to one of ordinary skill in the art.” Appellants respectfully submit that these assertions do not relate to the substance of Appellants’ arguments. More specifically, whether or not one of skill in the art was aware of “improving accessibility for data in a file system” is irrelevant to how that knowledge would somehow provide a reason for one to make the specific proposed combination. Such a broad reasons to combine can apply to numerous combinations in the prior art and do not

address the specific situation at hand. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In the present Advisory Action, the Examiner asserts “It should be noted that the combination was made so that the production database would be available during the load.” Appellants respectfully submit that making a combination for the sole purpose of teaching a claimed limitation is the definition of hindsight reasoning. Additionally, adding a reference only to add a specific teaching is not a reason to combine the references in the first place. The Examiner reiterates that improving accessibility for a file system data is a motivation; however, this motivation only identifies the presumed result of the combination and does not identify any particular reason for one of skill in the art to combine the references. Thus, the Examiner’s provided reason to combine the references is **improper** and based on hindsight reasoning. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response, the Examiner asserts that reconstruction is proper if based only on the knowledge of one of skill in the art. However, as already asserted, the Examiner’s reason to combine specifically recites one of Appellants’ claimed limitations “wherein the production database is available for access by users during said load” and there is no indication that one of skill in the art would be aware of Appellants’ limitation (nor has the Examiner found such a specific indication) before it was presented. The Examiner cannot simply cite a feature of Appellants’ claims as a reason for combining two disparate references and then assert that one of skill in the art would have that knowledge at the time. Therefore, the Examiner has failed to establish a *prima facie* rejection. Furthermore, as already argued, the combined references fail to teach or suggest all of Appellants’ claimed features.

## **Claims 2, 10, and 16**

**The Examiner has failed to establish a *prima facie* rejection because Kampe in view of Raman fails to teach or suggest wherein the refresh mechanism is further configured to perform post-processing on the database clone prior to said switch.**

With respect to this feature, the Examiner cites “(502-505)” without any further explanation. The cited portion of Kampe relates to creating a checkpoint and continuously updating the checkpoint in order to backup the component PC1. However, there is no indication of performing post-processing on a database clone of a production database prior to switching the database clone to be the production database. Instead, the cited steps relate to keeping a replica up to date (for failover purposes) via a checkpointing service (see paragraphs [0057]-[0060]). Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response to these arguments, the Examiner first asserts that continuously updating a checkpoint of a replica of a component teaches this feature since the processing happens after the replica is generated and before the failover switch. Appellants respectfully submit that the constant updating of a replica of a component in a clustered system does not relate to a database clone where new data is loaded into the database clone, post processing is performed on the database clone, and then the file system data is used as the file system data of the production database. As already argued maintaining a failover checkpoint does not relate to loading new data in a database clone, and correspondingly, keeping the checkpoint up to date (which still does not relate to new data) fails to teach or suggest performing post processing on such a database clone. Furthermore, the Examiner fails to provide any specific citations in the prior art for “there must be further post-processing on the database clone in the form of any preparation commands before the switch.” Therefore, the Examiner has failed to establish a *prima facie* rejection.

### **Claims 5, 13, and 19**

**Kampe in view of Raman fails to teach or suggest wherein the generated database clone includes references to data in the production database.**

With respect to this feature, the Examiner asserts:

Kampe as applied above uses a “checkpoint service” and “replica,” which must include references to data in the production database or else the data could not be restored to the primary component (see above and para. 0065).

Appellant respectfully submits that the Examiner is simply incorrect. The checkpoint service and replica does not necessarily include references to data in the primary component (which is not a production database). For example, it is quite likely that the checkpoint service simply copies the data to the replica, and the replica does not include references to data in the production database. In fact, since Kampe is directed to maintaining these replicas for failover purposes (which is clearly not the same as loading new data into a database clone, as already argued), it would not make sense to keep references to the data that should be backed up. In such a case, it would be very likely that in the event of a failover of the primary component, the replica would not be usable for restoring the primary component. Moreover, the cited portion [0065] (502-505 already addressed above) relates to failover of the primary component PC1 and retrieval of data from the checkpoint and makes no mention of a database clone which includes references to data in the production database. Thus, the cited portion of Kampe fails to teach or suggest this feature of the claims. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response, the Examiner asserts that “there is a relation between data from the replica and the data in the primary component” and between “the replica itself and the primary component”. The Examiner goes on to assert “if there were not references, then the primary component could not be reliably restored with data from the replica, and the system would not know that a particular replica will be used for restoring a particular primary component”. Appellants respectfully submit that the Examiner has clearly improperly interpreted the clear language of the claim. The claim specifically requires that the generated database clone includes “references to data in the production

database". Thus, without the data in the production database (e.g., which occurs when, using the Examiner's flawed interpretation, the failover occurs), the database clone would not be usable as it would refer to data that no longer exists. As one of skill in the art understands, **it does not make any sense for a replica to include references to data of the very thing it backs up** (still referring to the Examiner's flawed use of Kampe's failover). Such an interpretation of Kampe (which is not supported by any portion of Kampe) is simply **incorrect**. Therefore, the Examiner has failed to establish a *prima facie* rejection.

#### **Fourth Ground of Rejection**

Claims 3, 11 and 17 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Ishihara et al. (U.S. Patent 6,636,876) (hereinafter "Ishihara"). Appellant respectfully traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

##### **Claims 3, 11, and 17**

**1. The Examiner has failed to establish a *prima facie* rejection because Kampe in view of Raman and Ishihara fails to teach or suggest stopping the production database prior to said switching; and starting the production database after said switching.**

With respect to this feature, the Examiner relies on column 6, lines 27-50 of Ishihara. The cited portion teaches the stopping of a first database and the starting of a second duplicate database in order to perform maintenance on the hardware supporting the first database. Appellants submit that stopping a first database and starting a second database is not the same as stopping the first production database, switching the file system to the storage checkpoint (to be the file system data for the production database) and

restarting **the same production database**. Instead, Ishihara teaches starting a different database. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response to these arguments, the Examiner asserts that the second database becomes the production database. Appellants respectfully submits that these claims specifically recite that storage checkpoint becomes the file system data of **the** production database (which is clearly recited). Thus, it as the claims recite, it is **the same production database**. The Examiner cannot simply reinterpret the language clearly recited in the claims. Finally, the Examiner's assertions regarding "the use of the word 'switch'" implying that the roles of the production database and the other database also switch is not supported by the claims. The term "switch" in the claim clearly refers to file system data of the production database and not to the production database and database clone. The Examiner's assertions are clearly unsupported by the language used in the claims. Therefore, the Examiner has failed to establish a *prima facie* rejection.

**2. The Examiner has further failed to establish a *prima facie* rejection because the Examiner fails to provide a proper reason to combine the references.**

With respect to the combination of Kampe, Raman, and Ishihara, the Examiner asserts:

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kampe and Raman, such that "stopping the production database prior to the switch and starting the production database after the switch" is implemented. The motivation would have been to facilitate gracefully switching from one data system to another, as known to one of ordinary skill in the art.

Appellants respectfully submit that the Examiner has not provided any reason to make **the specific combination** and instead has merely provided a summary of the cited portion of Ishihara, which describes stopping a first database and starting a second duplicate database (with no new data) in order to perform maintenance on the hardware supporting the first database. There is no indication in this section (or in the Examiner's provided motivation) as to why (or how) stopping a first database and starting a second (duplicate)

database applies to the “primary component” of Kampe. More specifically, the cited portion does not relate to stopping a database, switching the storage checkpoint to be the file system data of the production database, and starting the production database, and instead, involves starting a different database. Thus, the cited portion and provided motivation does not give a reason to make the specific combination, and is thus **improper**. Therefore, the Examiner has failed to establish a *prima facie* rejection.

In response to these arguments, the Examiner simply reiterates his above assertions and then explains the correlations between the databases of Ishihara and the components of Kampe. Appellants did not have trouble understanding the Examiner’s combination, but instead assert that the provided reason to combine does not address the specific proposed combination. Additionally, the Examiner’s new motivation “to facilitate gracefully switching from one data system to another” also fails to address the specific combination and is no more than a broadly worded variation of Appellants’ claimed feature. Appellants’ features cannot be used as a reason to combine prior art references. Therefore, the Examiner has failed to establish a *prima facie* rejection.

### **Fifth Ground of Rejection**

Claims 4, 12 and 18 stand rejected as being unpatentable over Kampe in view of Raman and further in view of Appellants’ Admitted Prior Art (AAPA). Appellant respectfully traverses this rejection for at least the reasons given above regarding claims 1, 9, and 15.

## **CONCLUSION**

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-20 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-12400/RCK.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: December 8, 2008